

# Elastic search “容器化” 之路

分享人：陈刚

A central dark blue circle contains the title '目录' and 'CONTENT'. Five smaller circles are arranged around it, each containing a number and a topic. The circles are: 01 (top right), 02 (bottom right), 03 (bottom center), 04 (middle left), and 05 (top left). The 05 circle is the largest and has a diagonal line pattern. The 01 circle is also filled with a diagonal line pattern. The other circles are solid colors.

# 目录

## CONTENT

01

一切为了搜索  
All things for search

02

容器及容器编排  
Container and Container arrangement

03

ES的容器化  
Container of Elasticsearch

04

容器化之后  
After container

05

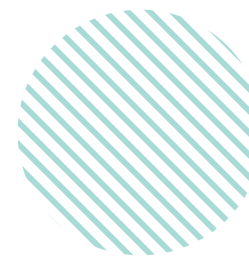
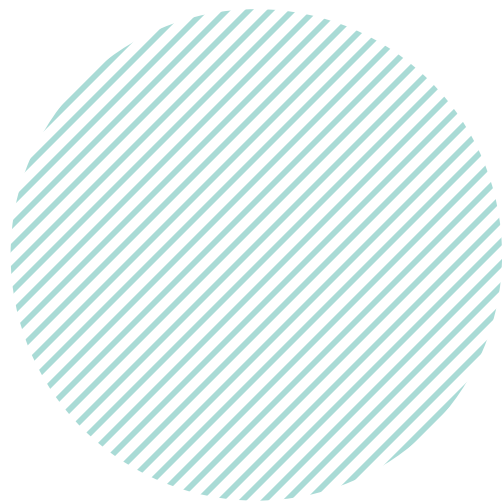
未来畅想  
In the feature



1

一切为了搜索

All things for search



## 为搜索所做的

### 分布式集群

cluster



### 易用接口

RESTful Interface

Elasticsearch Is Fast. Really, Really Fast.

01

02

03

04

### 倒排索引

Inverted index

### 分片和复制

shards & replicas



## 环境搭建

### **Environment construction**

Install Elasticsearch and modify this config file.

## 应用维护

### **Application maintenance**

There are many variations during using this application

## 集群管理

### **Cluster management**

You have to manage these many of nodes.

### 问题1

环境搭建及应用的配置需要  
耗费一定的工作量



### 问题4

不同索引数据之间如何做到资源隔离

### 问题2

Elastic的维护需要一定的人力

### 问题3

能否实现一键部署，自动化运维



# 2

## 容器及容器编排

Container and Container arrangement

# 容器



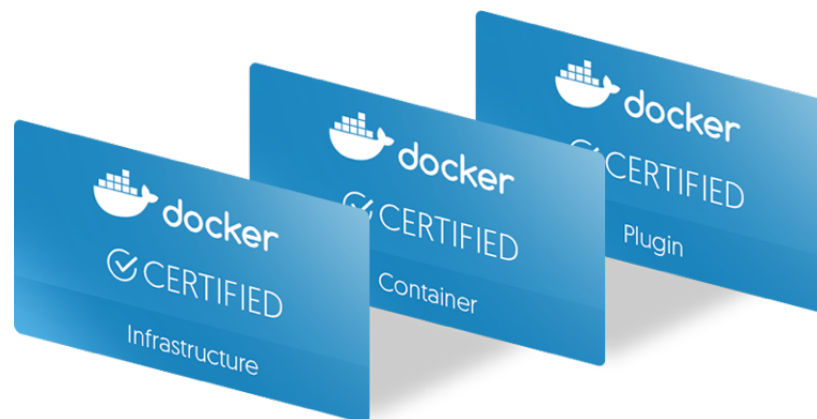
## About Containers

We have many PowerPoint templates that has been specifically designed.



## About Docker

Docker is the company driving the container movement and the only container platform provider to address every application across the hybrid cloud.





## 容器打包

容器通过Dockerfile 来自动编译生成相应的镜像

## 开发协作

容器通过Dockerfile 来自动编译生成相应的镜像

## 运维支持

容器通过容器编排工具实现可扩展

Information

2017



# 容器编排工具

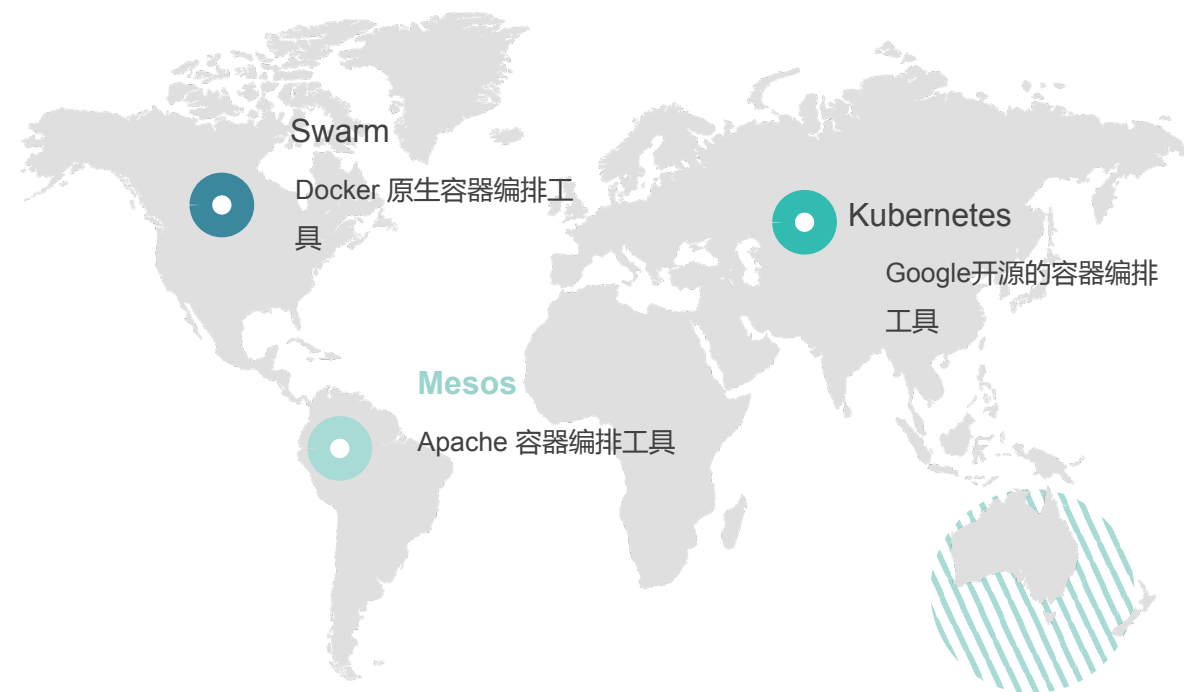


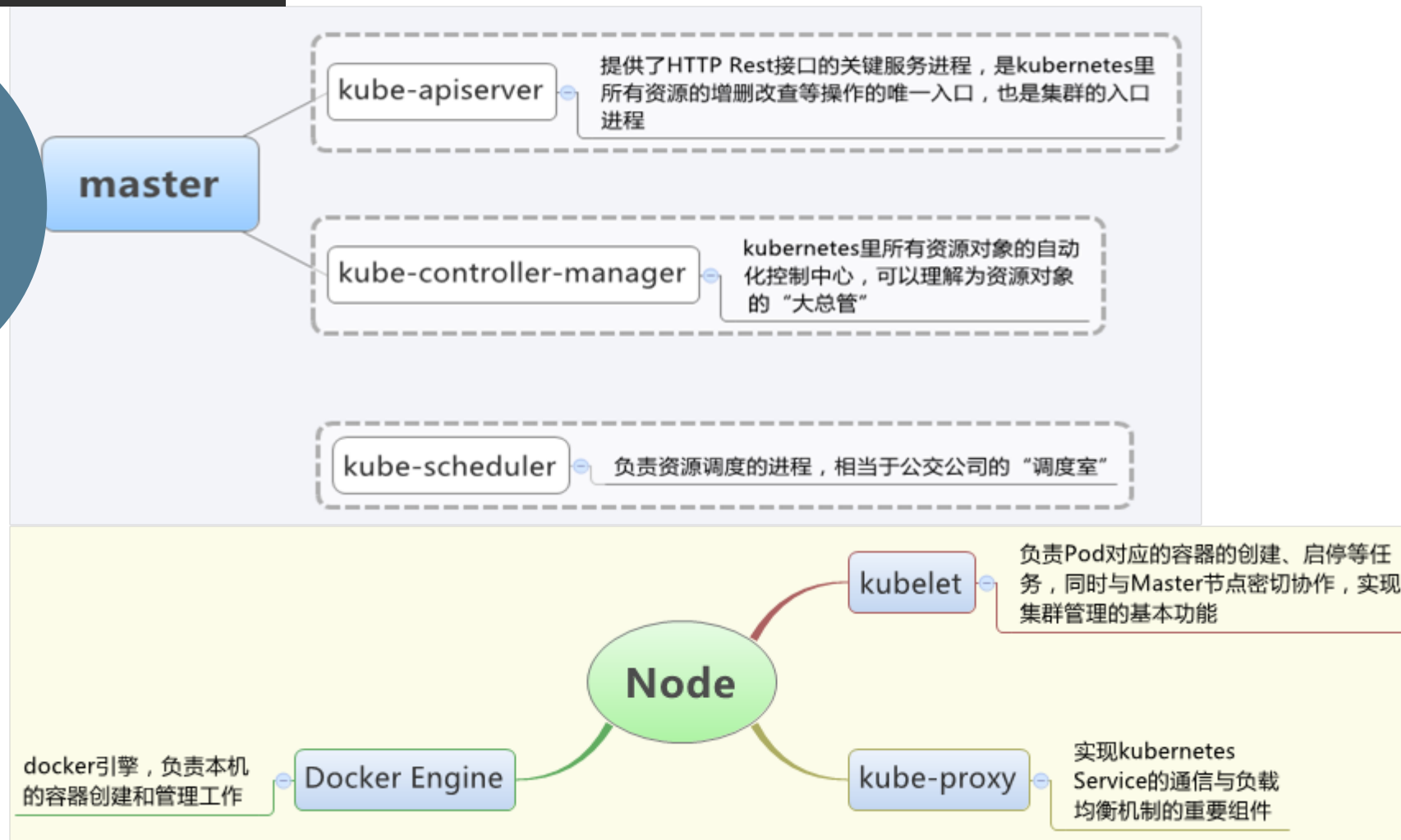
Information

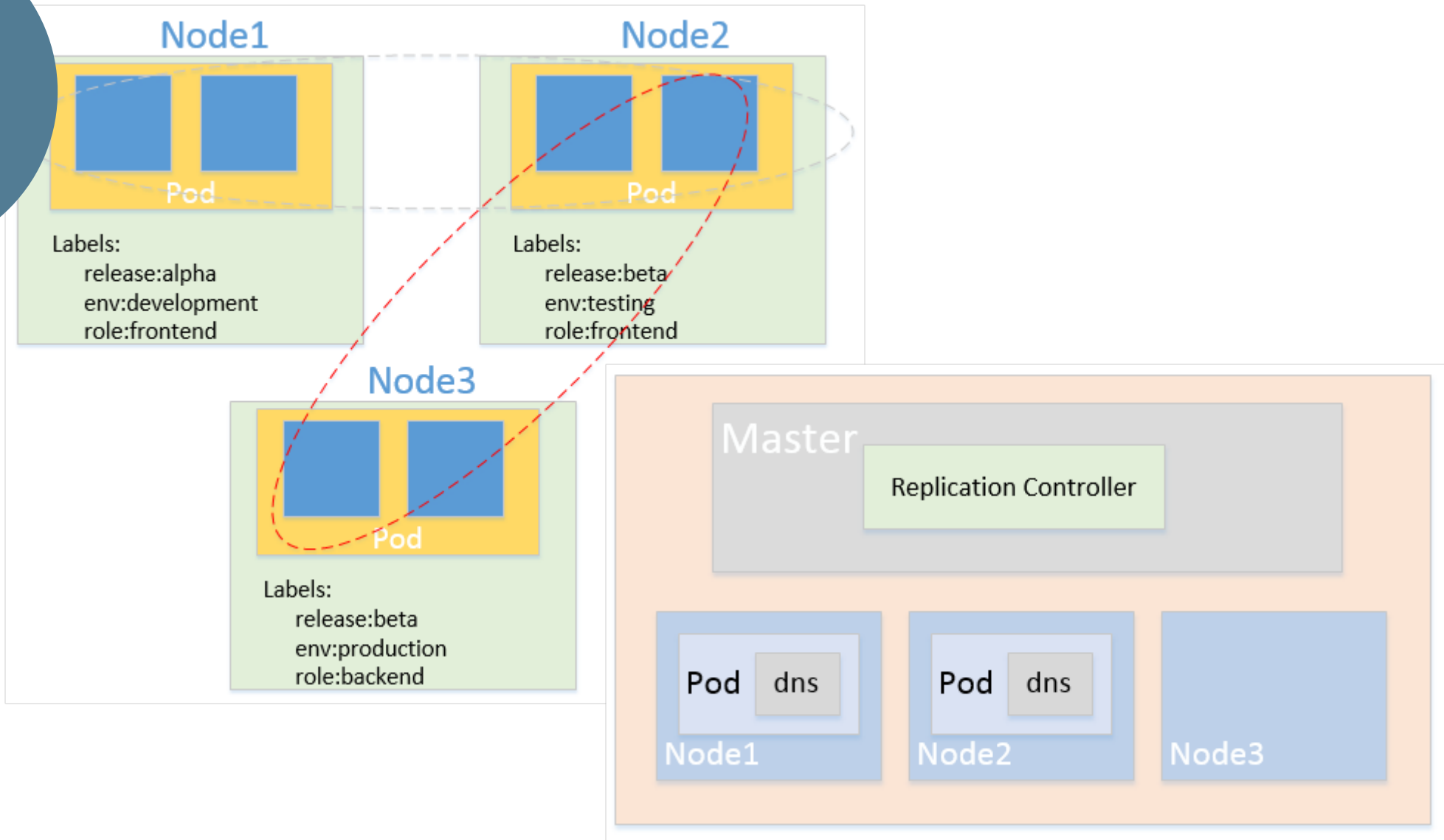
20111

# 容器编排工具

- Swarm
- Kubernetes
- Mesos





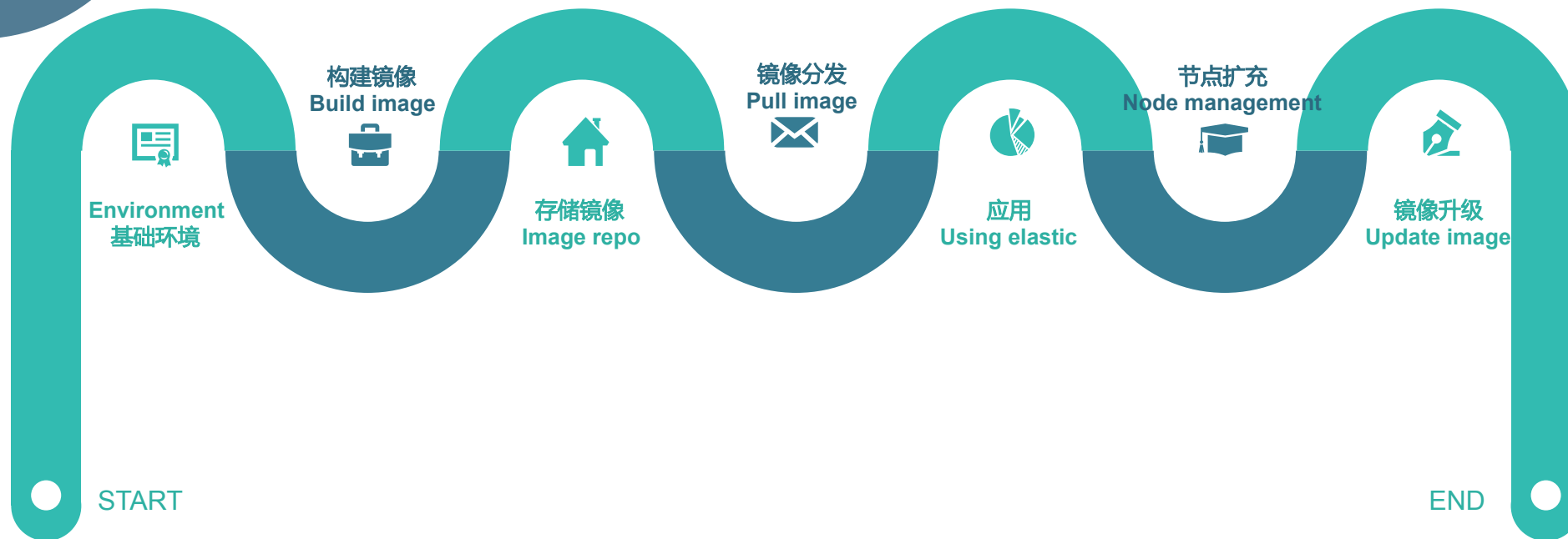




3

## ES的容器化

Container of Elasticsearch



# 容器化流程图

# 集群节点管理

## Node of cluster management



- Master nodes  
intended for clustering management only, no data, no HTTP API
- Client  
intended for client usage, no data, with HTTP API
- Data  
intended for storing and indexing data, no HTTP API

```
env:
- name: NAMESPACE
  valueFrom:
    fieldRef:
      fieldPath: metadata.namespace
- name: NODE_NAME
  valueFrom:
    fieldRef:
      fieldPath: metadata.name
- name: "CLUSTER_NAME"
  value: "myesdb"
- name: "NUMBER_OF_MASTERS"
  value: "2"
- name: NODE_MASTER
  value: "true"
- name: NODE_INGEST
  value: "false"
- name: NODE_DATA
  value: "false"
- name: HTTP_ENABLE
  value: "false"
- name: "ES_JAVA_OPTS"
  value: "-Xms256m -Xmx256m"
```

es-master.yaml

```
env:
- name: NAMESPACE
  valueFrom:
    fieldRef:
      fieldPath: metadata.namespace
- name: NODE_NAME
  valueFrom:
    fieldRef:
      fieldPath: metadata.name
- name: "CLUSTER_NAME"
  value: "myesdb"
- name: NODE_MASTER
  value: "false"
- name: NODE_INGEST
  value: "false"
- name: HTTP_ENABLE
  value: "false"
- name: "ES_JAVA_OPTS"
  value: "-Xms256m -Xmx256m"
```

es-data.yaml

```
env:
- name: NAMESPACE
  valueFrom:
    fieldRef:
      fieldPath: metadata.namespace
- name: NODE_NAME
  valueFrom:
    fieldRef:
      fieldPath: metadata.name
- name: "CLUSTER_NAME"
  value: "myesdb"
- name: NODE_MASTER
  value: "false"
- name: NODE_DATA
  value: "false"
- name: HTTP_ENABLE
  value: "true"
- name: "ES_JAVA_OPTS"
  value: "-Xms256m -Xmx256m"
- name: "NETWORK_HOST"
  value: "_site_,_lo_"
```

es-client.yaml





# 端口映射

Port mapping

- Http Port
  - 9200
- Tcp Port
  - 9300

```
ports:
- containerPort: 9300
  name: transport
  protocol: TCP
livenessProbe:
  tcpSocket:
    port: 9300
volumeMounts:
- name: storage
  mountPath: /data
volumes:
- emptyDir:
    medium: ""
  name: "storage"
```

es-master.yaml

```
ports:
- containerPort: 9300
  name: transport
  protocol: TCP
livenessProbe:
  tcpSocket:
    port: 9300
  initialDelaySeconds: 20
  periodSeconds: 10
volumeMounts:
- name: storage
  mountPath: /data
volumes:
- emptyDir:
    medium: ""
  name: "storage"
```

es-data.yaml

```
ports:
- containerPort: 9200
  name: http
  protocol: TCP
- containerPort: 9300
  name: transport
  protocol: TCP
livenessProbe:
  tcpSocket:
    port: 9300
readinessProbe:
  httpGet:
    path: /_cluster/health
    port: 9200
  initialDelaySeconds: 20
  timeoutSeconds: 5
volumeMounts:
- name: storage
  mountPath: /data
volumes:
- emptyDir:
    medium: ""
  name: "storage"
```

es-client.yaml

# 资源调度

## Resource scheduling

- Kubernetes Dashboard
- dashboard.yaml
- dashboardsvc.yaml

```
dashboard.yaml x
1  apiVersion: extensions/v1beta1
2  kind: Deployment
3  metadata:
4  # Keep the name in sync with image version and
5  # gce/coreos/kube-manifests/addons/dashboard counterparts
6  name: kubernetes-dashboard-latest
7  namespace: kube-system
8  spec:
9  replicas: 1
10 template:
11   metadata:
12     labels:
13       k8s-app: kubernetes-dashboard
14       version: latest
15       kubernetes.io/cluster-service: "true"
16   spec:
17     containers:
18     - name: kubernetes-dashboard
19       image: gcr.io/google_containers/kubernetes-dashboard-amd64:v1.5.1
20       resources:
21         # keep request = limit to keep this container in guaranteed class
22         limits:
23           cpu: 100m
24           memory: 50Mi
25         requests:
26           cpu: 100m
27           memory: 50Mi
28       ports:
29       - containerPort: 9090
30       args:
31       - --apiserver-host=http://10.0.251.148:8080
32       livenessProbe:
33         httpGet:
34           path: /
35           port: 9090
36         initialDelaySeconds: 30
37         timeoutSeconds: 30
```

```
dashboardsvc.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4  name: kubernetes-dashboard
5  namespace: kube-system
6  labels:
7  k8s-app: kubernetes-dashboard
8  kubernetes.io/cluster-service: "true"
9  spec:
10 selector:
11   k8s-app: kubernetes-dashboard
12 ports:
13 - port: 80
14   targetPort: 9090
```

Admin

Namespaces

Nodes

Persistent Volumes

Namespace

default

Workloads

Deployments

Replica Sets

Replication Controllers

Daemon Sets

Stateful Sets

Jobs

Pods

Services and discovery

Services

Ingresses

Storage

### Nodes

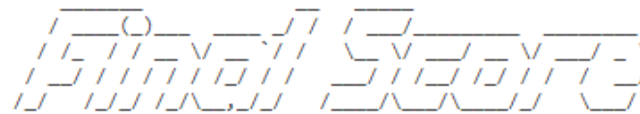
Name	Labels	Ready	Age
<a href="#">k8s-node-1</a>	<code>beta.kubernetes.io/arch: a...</code> <code>beta.kubernetes.io/os: linux</code> <code>kubernetes.io/hostname: k...</code>	True	9 days
<a href="#">k8s-node-2</a>	<code>beta.kubernetes.io/arch: a...</code> <code>beta.kubernetes.io/os: linux</code> <code>kubernetes.io/hostname: k...</code>	True	9 days



4

容器化后

After container



# 性能分析 相关因素

## Esrally

- pip3 install esrally
- esrally --distribution-version=5.2.2

Metric	Task	Value	Unit
Indexing time		28.0997	min
Merge time		6.84378	min
Refresh time		3.06045	min
Flush time		0.106517	min
Merge throttle time		1.28193	min
Median CPU usage		471.6	%
Total Young Gen GC		16.237	s
Total Old Gen GC		1.796	s
Index size		2.60124	GB
Totally written		11.8144	GB
Heap used for segments		14.7326	MB
Heap used for doc values		0.115917	MB
Heap used for terms		13.3203	MB
Heap used for norms		0.0734253	MB
Heap used for points		0.5793	MB
Heap used for stored fields		0.643608	MB
Segment count		97	
Min Throughput	index-append	31925.2	docs/s
Median Throughput	index-append	39137.5	docs/s
Max Throughput	index-append	39633.6	docs/s
50.0th percentile latency	index-append	872.513	ms
90.0th percentile latency	index-append	1457.13	ms
99.0th percentile latency	index-append	1874.89	ms
100th percentile latency	index-append	2711.71	ms
50.0th percentile service time	index-append	872.513	ms
90.0th percentile service time	index-append	1457.13	ms
99.0th percentile service time	index-append	1874.89	ms
100th percentile service time	index-append	2711.71	ms
...	...	...	...
...	...	...	...
Min Throughput	painless_dynamic	2.53292	ops/s
Median Throughput	painless_dynamic	2.53813	ops/s
Max Throughput	painless_dynamic	2.54401	ops/s
50.0th percentile latency	painless_dynamic	172208	ms
90.0th percentile latency	painless_dynamic	310401	ms
99.0th percentile latency	painless_dynamic	341341	ms
99.9th percentile latency	painless_dynamic	344404	ms
100th percentile latency	painless_dynamic	344754	ms
50.0th percentile service time	painless_dynamic	393.02	ms
90.0th percentile service time	painless_dynamic	407.579	ms
99.0th percentile service time	painless_dynamic	430.806	ms
99.9th percentile service time	painless_dynamic	457.352	ms
100th percentile service time	painless_dynamic	459.474	ms



OUR SERVICES

Creative Project

Creative Project

Creative Project

容器化欠缺总结



5

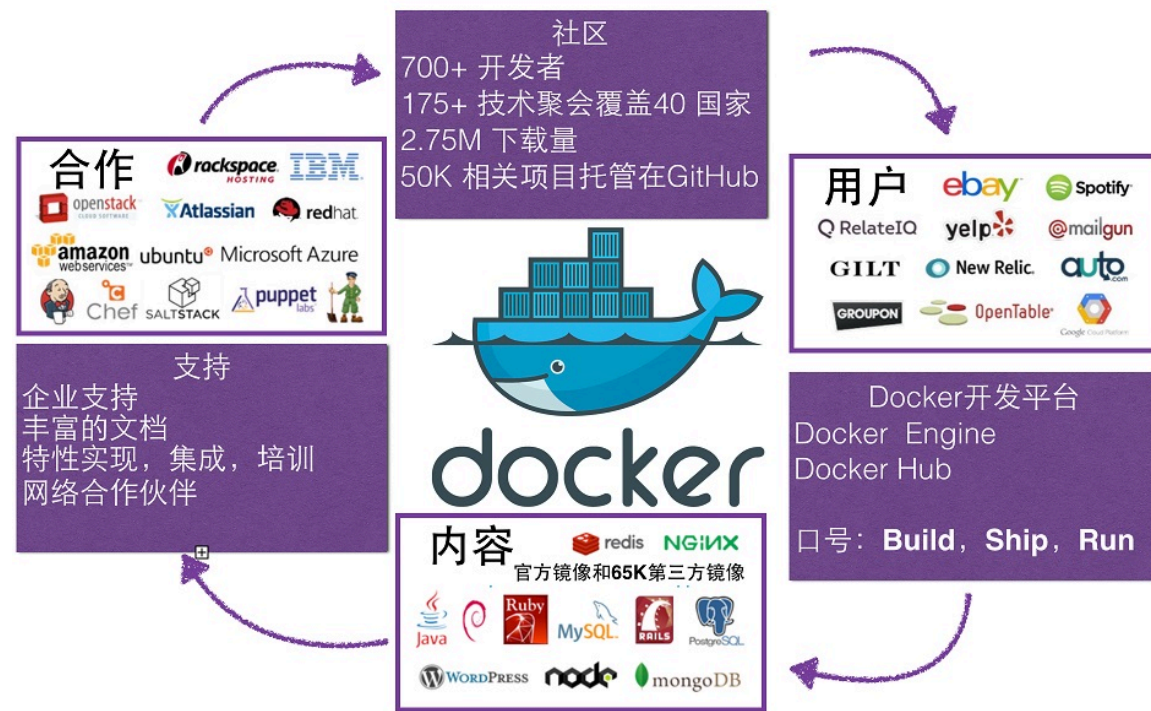
未来畅想

In the feature

# Docker生态圈

## 未来应用容器化

# Docker生态圈





## 未来应用趋势



应用集中化管理



应用性能更优



应用利于监控



应用更易扩展

# 未来已来

The past is gone and static. Nothing we can do will change it. The future is before us and dynamic.



**C h e n   G a n g**

[chengang\\_kzx@citicbank.com](mailto:chengang_kzx@citicbank.com)



14 NOVEMBER 2017 **RELEASES**

## Elasticsearch 6.0.0 GA released

By Clinton Gormley

Share



# 附录

Elasticsearch: <https://www.elastic.co/>

Docker: <https://www.docker.com/>

Elasticsearch-docker: <https://github.com/elastic/elasticsearch-docker>

Kubernetes: <https://kubernetes.io/>

Mesos: <https://mesos.apache.org/>

# 谢谢观看

THANKS FOR WATCH